

Integrated Systems Health Management for Intelligent Systems

Fernando Figueroa¹

NASA Stennis Space Center, MS, 3952, USA

and

Kevin Melcher²

NASA Glenn Research Center, Cleveland, OH, 44135, USA

Nomenclature

CSG	=	Chemical Steam Generator
CBM	=	Condition Based Maintenance
DIAK	=	Data, Information, and Knowledge
DIAKA	=	Data, Information, and Knowledge Architecture
FCL	=	Functional Capability Level
FMEA	=	Failure Modes and Effects Analysis
GN	=	Gaseous Nitrogen
IMBT	=	ISHM Model Building Toolkit
IPA	=	Isopropyl Alcohol
IS	=	Intelligent Sensor
ISHM	=	Integrated Systems Health Management
ISHM-DM	=	ISHM Domain Model
KSC	=	Kennedy Space Center
LC-20	=	Launch Complex 20
LOX	=	Liquid Oxygen
NASA	=	National Aeronautics and Space Administration
NCAP	=	Network Capable Application Processor
OSA-CBM	=	Open Systems Architecture for Condition-Based Maintenance
RCA	=	Root Cause Analysis
RETS	=	Rocket Engine Test Stand
S4	=	Systematic Sensor Selection Strategy
SS	=	Smart Sensor
S&As	=	Sensors and Actuators
SS&As	=	Smart Sensors and Actuators
SSC	=	Stennis Space Center
STE	=	Special Test Equipment
TCP/IP	=	Transmission Control Protocol/Internet Protocol
TEDS	=	Transducer Electronic Data Sheet
TIM	=	Transducer Interface Module
WISE	=	Virtual Intelligent Sensor Environment

I. Abstract

The implementation of an integrated system health management (ISHM) capability is fundamentally linked to the management of data, information, and knowledge (DIAK) with the purposeful objective of determining the health of a system. Management implies storage, distribution, sharing, maintenance, processing, reasoning, and

¹ Innovative Partnerships Division, Stennis Space Center, MS 39529, AIAA Associate Fellow.

² Senior Research Engineer, RHC/Controls & Dynamics Branch, 21000 Brookpark Road, Mail Stop 77-1, AIAA Senior Member.

presentation. ISHM is akin to having a team of experts who are all individually and collectively observing and analyzing a complex system, and communicating effectively with each other in order to arrive at an accurate and reliable assessment of its health. In this chapter, concepts, procedures, and approaches are presented as a foundation for implementing an ISHM capability relevant to intelligent systems. The capability stresses integration of DIaK from all elements of a system, emphasizing an advance toward an on-board, autonomous capability. Both ground-based and on-board ISHM capabilities are addressed. The information presented is the result of many years of research, development, and maturation of technologies, and of prototype implementations in operational systems.

II. Introduction

In this chapter, Integrated Systems Health Management (ISHM) is presented as an enabling discipline/technology area for intelligent systems, as well as a capability that embodies “intelligence” in itself. To that end, a variety of intelligent systems-relevant ISHM topics are addressed, including relevant examples. The information presented provides the reader with an understanding of the state-of-the-art, current research, and challenges that are relevant to ISHM as a core capability of an intelligent system.

ISHM has been defined from many perspectives. Here it is defined as a capability that is achieved by integrating data, information, and knowledge (DIaK) that is conceptually and/or physically distributed throughout the system elements (which inherently implies the capability to manage DIaK associated with distributed sub-systems). The term DIaK management encompasses contextual and timely storage, distribution, sharing, maintenance, processing, reasoning, and presentation. This paradigm implies that DIaK must be available to any element of a system at the right time and in accordance with a meaningful context. ISHM Functional Capability Level (FCL) is measured by how well a system performs the following functions: (1) detect anomalies, (2) diagnose causes, (3) predict future anomalies/failures, (4) enable efficient integration and execution of all phases of the life-cycle of a system from a systems engineering perspective, and (5) provide the user with an integrated awareness about the condition of important elements of the system as a means of guiding user decisions.

The paper is organized as follows: Section III describes core areas of ISHM capability development, including standards and related architectures, the ISHM knowledge model, software tools, intelligent sensors and components, and sensor selection and placement. Section IV describes ISHM in the context of systems design, integration, and engineering. Section V describes briefly controls for ISHM-Enabled systems. Section VI describes briefly opportunities for advances in validation and verification of ISHM systems. Section VII includes examples of some relevant implementations, and Section VIII provides conclusions and recommendations on the way forward.

III. ISHM Capability Development

ISHM functions are currently performed manually. For complex systems; it involves many people; it is very costly and difficult to improve with time and use; it involves minimal integration of DIaK across the system; it is not comprehensive (does not include all elements of a system or much DIaK about the system); and it is not continuous (a people-based system is generally not vigilant 24 hours a day, every day). Figure 1 describes the layered approach currently employed to achieve ISHM capability. At the top layer, Layer 1, on-board ISHM capability is deployed. At the moment, this amounts to monitoring thresholds on a few sensor measurements in order to avoid catastrophic events. Only events (anomalies) are detected, which are used by people in the lower layers to reason and infer what the associated anomaly might be; diagnose its causes, etc. Layers 2 to 4 involve people in increasing numbers, and even entire organizations aligned with individuals in the control/evaluation rooms. ISHM capability on-board the system is expected to enable faster and more accurate analysis, reasoning, and decision making in layers 2-4.

There are a number of implementations of ISHM capability since before 2000. NASA GRC led some advances in health management for propulsion systems [1]. These efforts included a combination of sensor validation methods and expert systems for diagnosis, applied to rocket engine test post diagnostics. Another example is the Boeing’s 777 Airplane Health Management (AHM) system [2]. AHM involves a central maintenance computer that collects information from many subsystems that encompass built-in test elements. AHM’s purpose was to decrease unplanned maintenance from the 75% level to a 25% level. This level of improvement could only be possible by augmenting the knowledge and information base in isolated subsystems, through processing and reasoning across subsystems. This strategy was supported by the AHM architecture. Boeing and Pratt & Whitney Rocketdyne implemented the Advanced Health Management System (AHMS) in the Space Shuttle Main Engine (SSME) [3]. AHMS was developed to meet more stringent engine reliability requirements. The long term goals were ambitious, encompassing an integrated approach to detect anomalies, diagnose causes, and predict future anomalies; however, only a first phase was implemented, encompassing monitoring of vibration sensors mounted on the high-pressure

fuel turbopump and high-pressure oxidizer turbopump. AHMS was certified and used in flight, with the authority to shut down the engine.

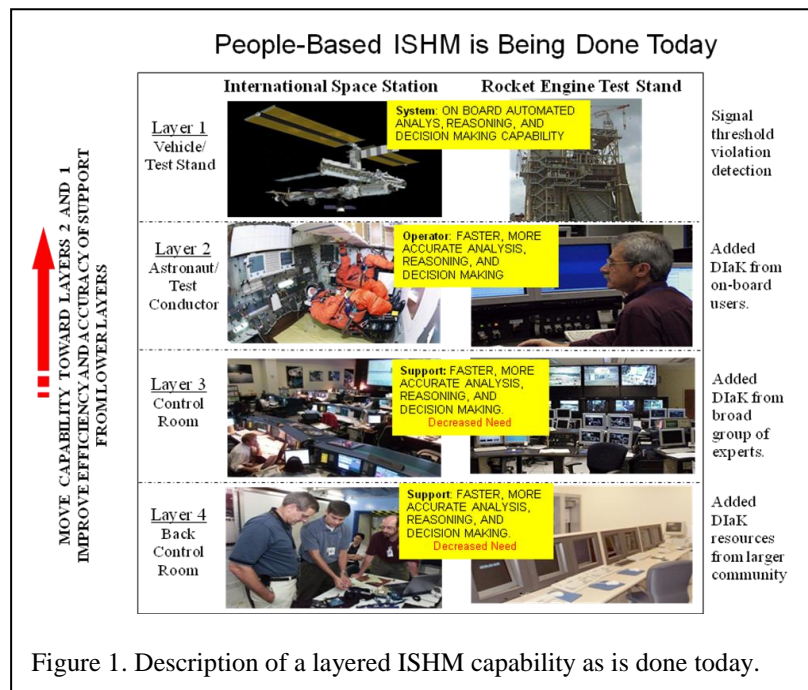
Another attempt to advance ISHM capability implementation was embodied by NASA's Propulsion IVHM Technology Experiment (PITEX), where IVHM stands for Integrated Vehicle Health Management. PITEX implemented an architecture that represented system elements with state models, and used states of system parameters to reason and make decisions about the health of the elements. PITEX used the software environment Livingstone to model a propulsion system composed of tanks, valves, and other basic elements; it was tested using simulated data [4].

A system that has been in use for a long time is the Health and Usage Monitoring System (HUMS) for helicopters (a web search will provide a large number of references). The HUMS monitors data from helicopter subsystems and processes it using a set of specialized algorithms. The resulting anomaly indicators and original data are used by experts to infer if critical elements might be trending toward failure. In this system, knowledge and its integrated interpretation is primarily done by people.

Although the references cited above implement ISHM to some level of capability, they do not represent "intelligent" implementations that would encompass embedded DIaK, nor embrace intelligent systems architectures, paradigms, or ontologies. The following sections describe technologies, tools, and infrastructure needed to achieve ISHM capability that is mainly on-board the system, affordable and evolutionary throughout the life of the system, integrates DIaK across the system, is continuous, and is comprehensive. In order to make this possible, it is necessary that the ISHM capability must incorporate a knowledge-based approach, and hence embody "intelligence."

A. Standards for ISHM Implementation

The development of an ISHM capability requires the use of models (knowledge) applied to information and data associated with various elements that make up a system. Here, the term "model" is used in the broadest sense as it may include qualitative (e.g. heuristics), analytic, statistical, fuzzy-logic, classic logic, artificial neural network and other types of models. Use of models is enabled by management of DIaK, encompassing storage, distribution, sharing, maintenance, processing, reasoning, and presentation. In order to make this possible in a generic manner, meaning not for a specific application; standards must be established so that DIaK can be managed in a plug&play



and interoperable manner, and for affordability.

Standards for ISHM must be at a high enough layer in the infrastructure so that they are largely independent of the physical (e.g. Ethernet) and transmission (e.g. TCP/IP) layers. Example standards for ISHM include the IEEE 1451

family of standards for smart sensors and actuators, the Open Systems Architecture for Condition-Based Maintenance (OSA-CBM) standard, and the Open systems Architecture for Enterprise (OSA-EAI) standard managed by the Machine Information Management Open Standards Alliance (MIMOSA). These standards are sufficiently abstracted so that they can be implemented as part of any physical or transmission architecture.

1. IEEE 1451 Family of Standards for Smart Sensors and Actuators (SS&A)

The IEEE 1451 family of standards was developed by government and private entities under the leadership of the National Institute of Standards and Technology (NIST). Reference [5] provides a summary of the standards and their use. In creating these standards, the objective was to standardize DIaK associated with sensors and actuators (S&As). The standards are described as a family because, as evidenced from the quote in the following paragraph, they address various elements and functions of Smart Sensors and Actuators (SS&As). The notion is that SS&As must incorporate DIaK related to their functionality and provide their DIaK, via a communications network, to other systems or functions that use and manage S&As.

“The IEEE (Institute of Electrical and Electronics Engineers) 1451 smart transducer interface standards provide the common interface and enabling technology for the connectivity of transducers to microprocessors, control and field networks, and data acquisition and instrumentation systems. The standardized TEDS specified by IEEE 1451.2 allows the self-description of sensors and the interfaces provide a standardized mechanism to facilitate the plug and play of sensors to networks. The network-independent smart transducer object model defined by IEEE 1451.1 allows sensor manufacturers to support multiple networks and protocols. Thus, transducer-to-network interoperability is on the horizon. The inclusion of P1451.3 and P1451.4 to the family of 1451 standards will meet the needs of the analog transducer users for high-speed applications. In the long run, transducer vendors and users, system integrators and network providers can all benefit from the IEEE 1451 interface standards [6].”

The most common physical architectures for systems are bus-based multi-drop configurations. Figure 2 shows configurations and implementation of IEEE 1451 standards, and a short summary is provided below. The standards are still being modified, but the intent here is to provide a sense of how the standards can be used to enable interoperability and plug&play capability with networked transducers encompassing embedded information (hence, smart transducers).

IEEE P1451.0 defines a set of common commands, common operations, and Transducer Electronic Data Sheet (TEDS) for the family of IEEE 1451 standards. The commands allow communication with sensors or actuators in IEEE 1451-based wired and wireless networks. The functionality is independent of the physical communications media and the network node called Network Capable Application Processor (NCAP).

IEEE 1451.1 defines a common object model describing the behavior of smart transducers (sensor and actuators). It defines the communication models used for the standard, which include the client-server and publish-subscribe models. Application software based on IEEE 1451; running in the NCAP, communicate with transducers through any physical layer standards as needed for a particular application. The standard enables communications among NCAPs and to higher level systems, in a network neutral manner.

IEEE 1451.2 defines transducers-to-NCAP interface and TEDS for a point-to-point configuration. Transducers are part of a Transducer Interface Module (TIM). The original standard describes a communication layer based on enhanced SPI (Serial Peripheral Interface) with additional HW lines for flow control and timing.

IEEE 1451.3 defines a transducer-to-NCAP interface and TEDS for multi-drop transducers using a distributed communications architecture. It allows many transducers to be arrayed as nodes, on a multi-drop transducer network, sharing a common pair of wires.

IEEE 1451.4 defines a mixed-mode interface for analog transducers with analog and digital operating modes. A TEDS was added to a traditional two-wire, constant current excited sensor containing a FET amplifier. The TEDS model was also refined to include critical information that must fit in a small memory device, needed by very small transducers. Templates are used to describe the data structure of TEDS. The current templates cover accelerometers, strain gages, current loop sensors, microphones, thermocouples, and others.

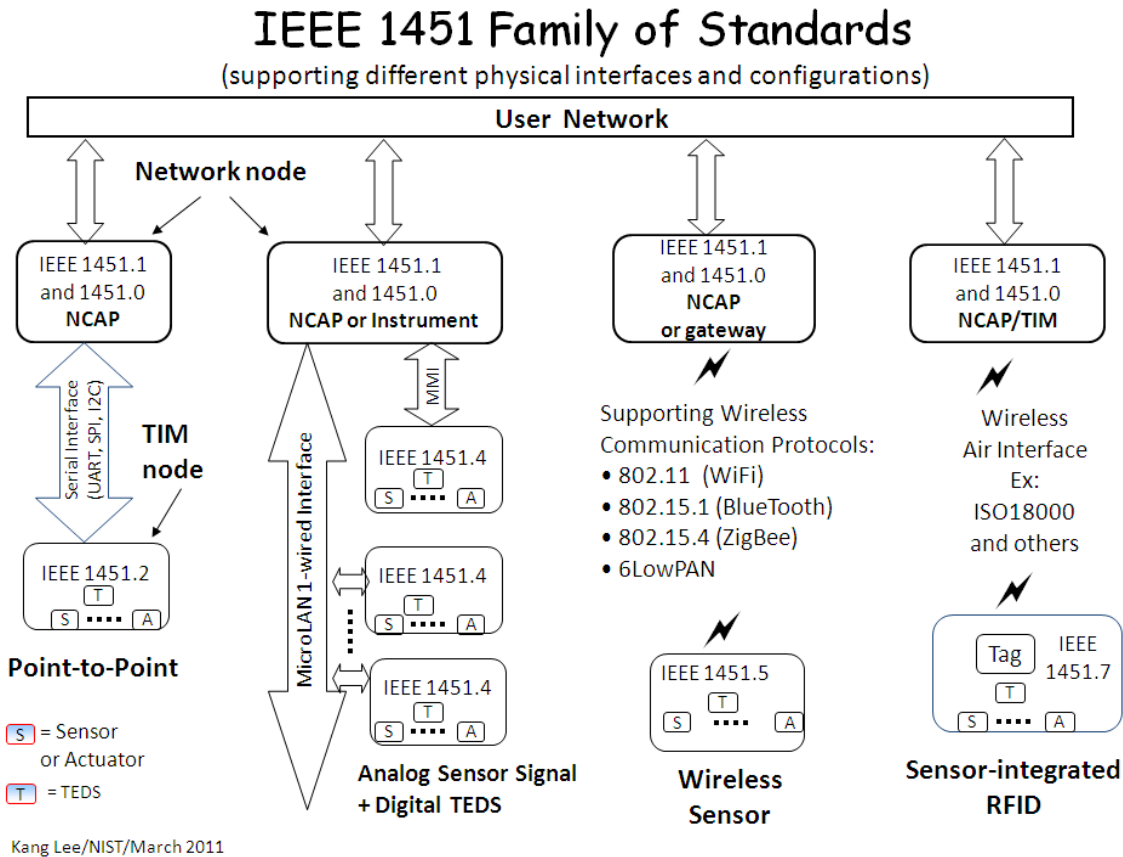


Figure 2. IEEE 1451 Family of Standards (Courtesy of Dr. Kang Lee from NIST).

IEEE P1451.5 defines a transducer-to-NCAP interface and TEDS for wireless transducers. Wireless communication protocol standards such as 802.11 (WiFi), 802.15.1 (Bluetooth), 802.15.4 (ZigBee) are being considered as some of the physical interfaces for IEEE P1451.5. The objective is to be able to communicate with a wireless transducer embodying any of these three wireless protocols.

IEEE P1451.6 defines a transducer-to-NCAP interface and TEDS using the high-speed CANopen network interface. Both intrinsically safe and non-intrinsically safe applications are supported. It defines a mapping of the 1451 TEDS to the CANopen dictionary entries as well as communication messages, process data, configuration parameter, and diagnosis information. It adopts the CANopen device profile for measuring devices and closed-loop controllers.

2. OSA-CBM Standard

The OSA-CBM standard was developed by government and private entities. This standard addresses management of health information from any element, subsystem, system, or system-of-systems. The foundation is the definition of layers where health information is organized according to the degree of processing, and hence amount of DIaK employed, to determine health condition (Figure 3) [7, 8]. The standard focuses on automated real-time management of health information. In contrast, health management over extended periods of time (non-real time) is typically based on large databases, and done primarily by people. The non real-time approach is standardized as the Open Systems Architecture for Enterprise Application Integration (OSA-EAI) Standard. Both standards are maintained by the Machine Information Management Open Standards Alliance (MIMOSA) organization [7].

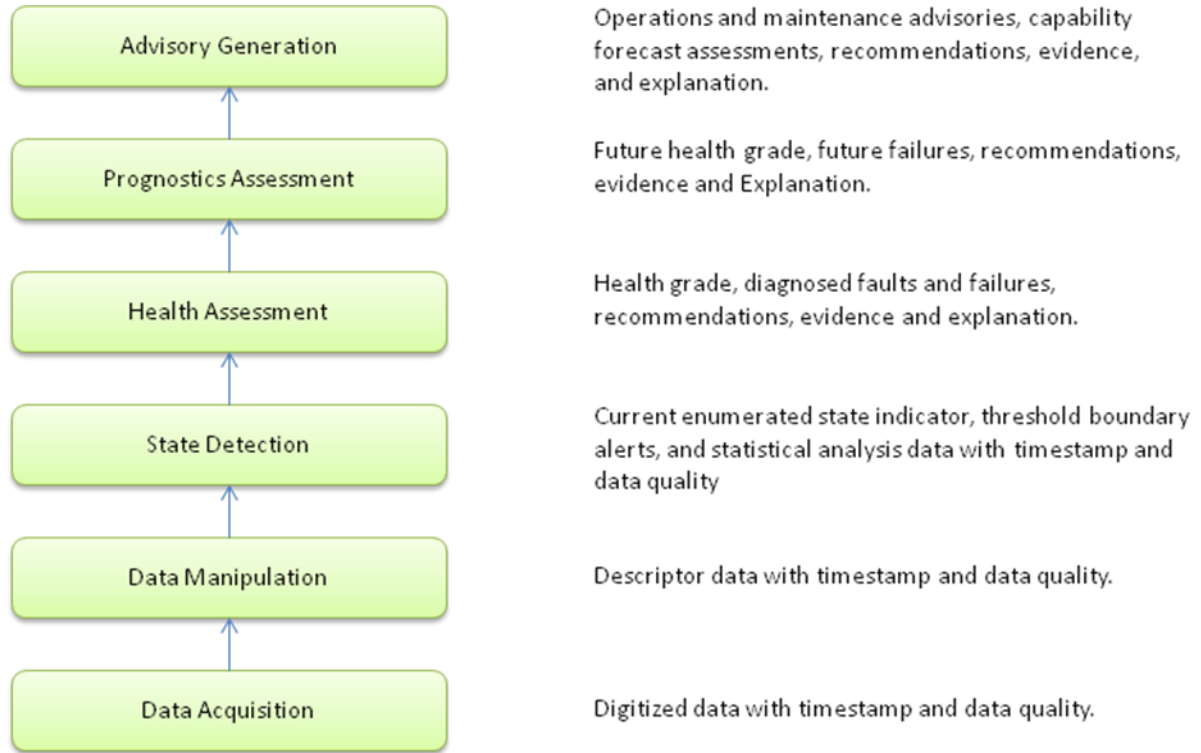


Figure 3. CBM layered health DIaK architecture.

3. Machine Information Management Open Standards Alliance (MIMOSA)

MIMOSA “is a non-profit trade association dedicated to developing and encouraging the adoption of open information standards for Operations and Maintenance in manufacturing, fleet, and facility environments. MIMOSA’s open standards enable collaborative asset lifecycle management in both commercial and military applications.” [7].

4. Example Implementation of IEEE 1451 and OSA-CBM Standards

Figure 4 shows a physical architecture (bus-based, multi drop, Ethernet network) for a pilot ISHM system implemented at NASA Kennedy Space Center, Launch Complex 20 (LC-20) [8]. The architecture is hierarchical, with buses at various levels, where higher-level information flows up toward the site-wide management computer. This is a typical architecture for systems in most industries, including aerospace. Standards were implemented in the lower part of the physical architecture (bus showing IEEE 1451.1 and OSA-CBM standards on Ethernet), but it was sufficient to demonstrate the impact of standards for ISHM implementation in an operational system, during a test at the LC-20.

The experiment demonstrated interoperability of ISHM systems developed by three different providers: NASA Stennis Space Center (NASA SSC), NASA Kennedy Space Center (NASA KSC), and the Pennsylvania State University’s Applied Research Laboratory (PSU-ARL). The interoperability was enabled through the use of the IEEE 1451 and OSA-CBM standards.

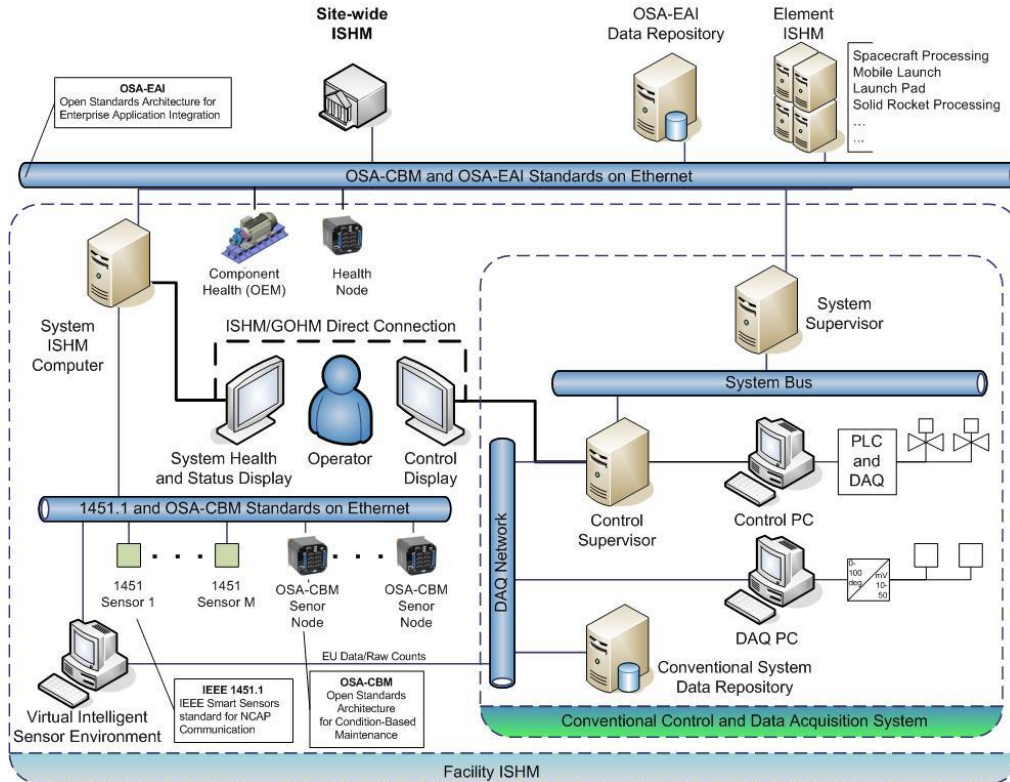


Figure 4. Architecture for pilot ISHM system implemented at NASA Kennedy Space Center, Launch Complex 20 (LC-20) showing the use of IEEE 1451.1, OSA-CBM, and OSA-EAI standards.

B. ISHM Knowledge Model (ISHM-DM)

The concept of an ISHM Domain Model (ISHM-DM) has been introduced previously [9, 10]. ISHM-DM embodies DIaK that is needed to achieve ISHM capability; including system element identification and specifications, and inter-element relationships used in reasoning approaches. Data is available from sensors and components. Distribution of DIaK associated among the physical elements of a system gives rise to an ISHM architecture that enables distributed management of DIaK to achieve ISHM functionality. The ISHM architecture is a DIaK Architecture (DIaKA), where intelligent processes (e.g. physics-based models) providing various degrees of integration (through inter-dependencies) are used to achieve the desired ISHM capability; and where DIaK are managed in a distributed manner (Figure 5). This hierarchical architecture enables abstracting models of processes occurring throughout the system (e.g. tank pressurization, subsystem leak, valve leak, sensor flat, etc.), and is conceptually different from a typical architecture depicting the physical composition of a system, where the hierarchy is based on simpler physical elements being assembled into more complex sub-systems and systems.

Figure 6 is another depiction of the DIaKA. DIaK are distributed among the elements of a system, including sensors, actuators, and components; as well as subsystems, and systems. The icons represent active repositories of data and information pertinent to their function, operation, and health. The icons also represent process models (knowledge) that enable ISHM functionality. Figure 6 shows a representation of the DIaKA as it is related to an ISHM-DM of a simple rocket test stand system.

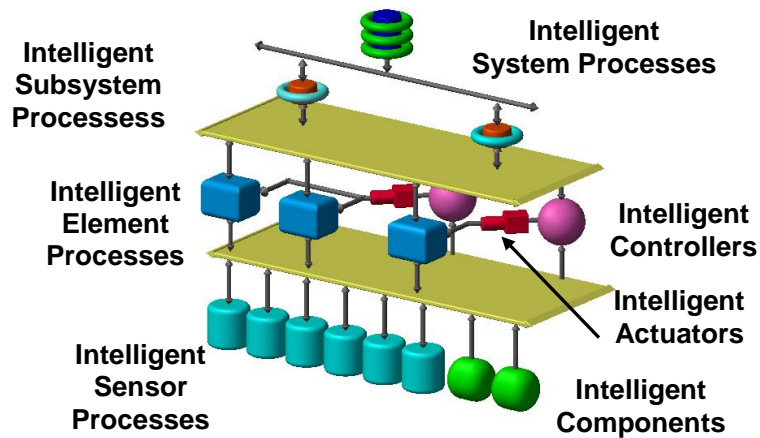


Figure 5. Abstract representation showing that Data, Information, and Knowledge Architecture is both distributed and hierarchical in an intelligent ISHM system.

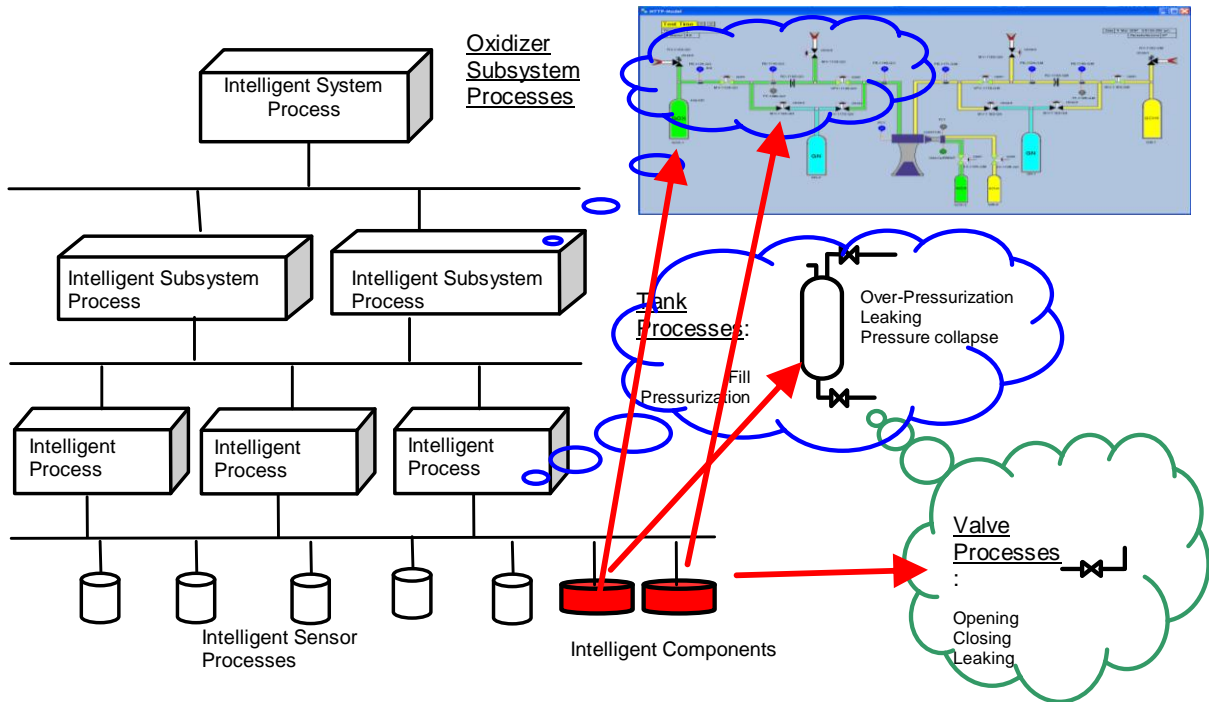


Figure 6. DIaKA showing the correspondence of practical ISHM-DM elements adapted for a small rocket test stand. Process models are executed often in parallel for consistency checking that leads to anomaly detection and reasoning about health of processes, sensors, and components.

The DIaKA supports the following paradigm. Sensor icons are repositories for sensor processes that operate on measurements within a local context, independent of other elements of the system. Sensor processes are, for example, algorithms to determine level of noise, changes on level of noise, flat signals, time response characteristics, etc. In addition, sensor processes include health assessment processes focused on determining: sensor health, and the quality of the measurement. Health assessment sensor processes also receive information from other processes higher in the hierarchy to improve their health assessment. For example, a process model of flow from a tank through a valve, to atmosphere; can allow consistency checks among pressure and temperature sensors along the

path of the flow. If one sensor is inconsistent with the model, this information is fed back to the sensor to improve its own health assessment and anomaly determination. The same applies to component processes. The final objective is to determine the health of sensors and components; and do it with maximum utilization of DIaK embodied in the various layers of processes in the hierarchy. This approach is described in reference [9].

C. Software Capabilities to Develop ISHM Domain Models

Core capabilities of ISHM include: (1) detect anomalies, (2) diagnose causes, (3) predict future anomalies/failures, (4) enable efficient integration and execution of all phases of the life-cycle of a system from a systems engineering perspective, and (5) provide the user with an integrated awareness about the condition of important elements of the system as a means of guiding user decisions. These capabilities are to provide continuous and comprehensive awareness about the health of every element of a system. DIaK must be employed to do the reasoning leading to achieving the core capabilities. Furthermore, multiple simultaneous process models and approaches should be employed to achieve maximum functional capability level (FCL), that is to make effective use of all DIaK embodied in the ISHM-DM. A software system for ISHM capability should support all core capabilities by integrating systematically DIaK through the ISHM-DM. The following requirements should be met by the software system:

Object representation: object representation of system physical elements and associated process models is the best way to embed DIaK in a systematic and in an organized manner. Object orientation also embodies re-use of software that is modularized into objects, and allows a more intuitive understanding of the code and its outcomes.

Distribution of ISHM-DM's within and across networks: ISHM-DM's might be distributed among processors connected to a network, simply because it is necessary to use parallel processing, and/or ISHM-DM's might be created by different people in various geographic locations. As complexity of systems increase, and/or a large number of process models are used in achieving effective ISHM capability, it is not reasonable or manageable to do this with a centralized architecture.

Distribution across processing units: Since multiple process models are expected to be running at any given time, the software environments should support parallel processing.

Inference engine: Many tasks require an inference engine. Reasoning and decision making leading to anomaly detection, diagnostics, effects, and prognostics; require contextual integrity and cause-effect analysis using heterogeneous data and information. The inference engine must also allow accurate representation and automatic execution of failure modes and effects analysis (FMEA).

Integrated management of distributed DIaK: DIaK must be managed in a way to allow embodiment of systems thinking across elements and subsystems. Often this is enabled by definitions of relationships among elements of systems that can be physically visible (i.e. attached to, belong to a system); or more abstracted relationships, as it relates to involvement in process models (e.g. pressure sensors associated to a particular subsystem, subsystem definitions that change with configuration, etc.).

Definition of dynamic relationships among objects for use in reasoning: Often, the framework for reasoning and application of process models changes dynamically with configuration changes, stages of operation, etc. This also means that relationships among objects and processes change dynamically, and must be represented in the ISHM-DM's. For example, reasoning to detect leaks in a sealed subsystem requires that membership of elements to sealed subsystems must change with valve state changes.

Iconic representation of systems objects with visible and virtual links (relationships) used to provide intuitive representation of reasoning and context: The mix of object representation and iconic representation of DIaK provides the ability to intuitively visualize interrelationships and dig deep into details of the ISHM system. As complexity increases, graphical programming and visualization become essential.

A software environment developed by NASA Stennis Space Center and General Atomics [9-13] meets all of the requirements above. The software was developed using G2 [14], which is a commercial programming environment for implementation of intelligent applications. Other software environments for creation of knowledge models include TEAMS [25] and MADe [26] for automating failure modes and effects analysis (FMEA), and Livingston for state-machine models [24].

D. Intelligent Sensors and Components

The lower elements in the DIaKA (Figure 6) represent processes associated with sensors and components; where "components" is intended to encompass any element that is not a sensor; e.g. tanks, pumps, etc. These elements directly represent physical entities in the system; and, in the future, they are expected to incorporate their own embedded processing and networking capabilities. This is already true for sensors, as many "intelligent sensor" concepts are now available commercially, and more are in development [15, 16].

There are many definitions for “Intelligent Sensor” or IS. The following definition is based on the foundation provided by the IEEE 1451 family of standards for Smart Sensors and Actuators. It is reasonable to assume that the standard defines a “Smart Sensor (SS),” as described previously in Section A “Standards for ISHM Implementation.” “Intelligent Sensor” is therefore a “Smart Sensor” with the ability to provide the following functionality: (1) measurement, (2) assessment of the quality of the measurement, and (3) determination of the “health” of the sensor. The better the sensor provides functionalities 2 and 3, the more intelligent it is.

Implementation of IS’s can be done in many ways. Commercial SS incorporating TEDS in-a-chip have been available for some time (a web search will reveal many offerings). Some IS or SS modules have been developed in industry. These are small format units that incorporate signal conditioning, data acquisition, processing capability, and protocols for communicating as network elements [15, 16]. In other cases, IS capability is enabled by a combination of hardware and software that turns classic sensors into smart sensors; as is the case with products from National Instruments [17]. Intelligent sensor functionality has also been implemented purely in software, again, to turn classic sensors into intelligent ones. Figure 7 shows the configuration of a pilot ISHM implementation for a rocket engine test stand. Here the Virtual Intelligent Sensor Environment (VISE) turns all classic test-stand sensors into intelligent sensors [10]. The VISE publishes IS data and information to a bus for consumption by the ISHM system and other users such as repositories and visualization systems. Some of the processes to be embedded in IS’s include:

- Noise Level Assessment and History
- Spike Detection and History
- Flat Signal Detection and History
- Response Time Characterization
- Intermittency Characterization and History
- Physical Detachment Characterization and History
- Regime Characterization and History
- Curve Fit on Identified Regimes

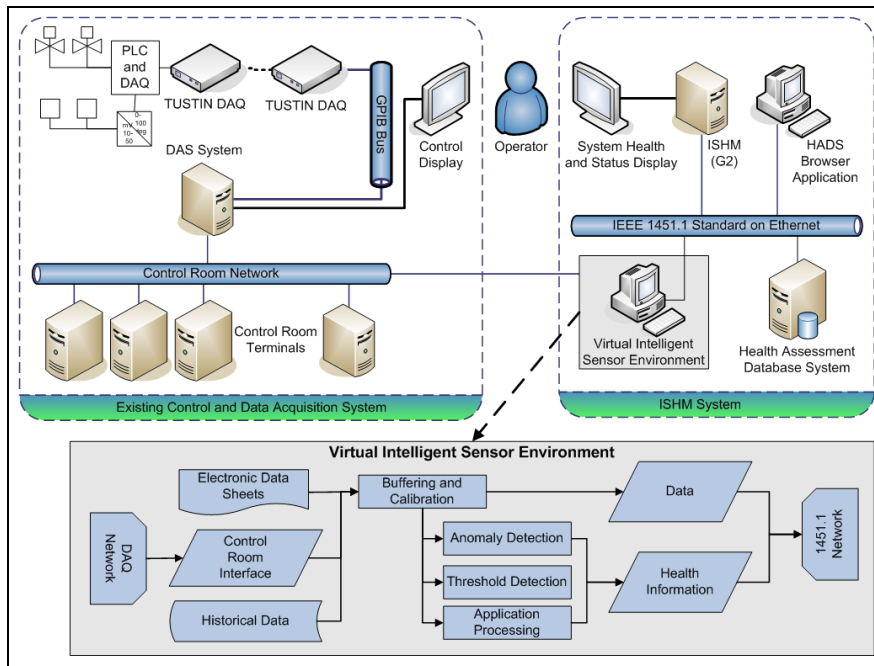
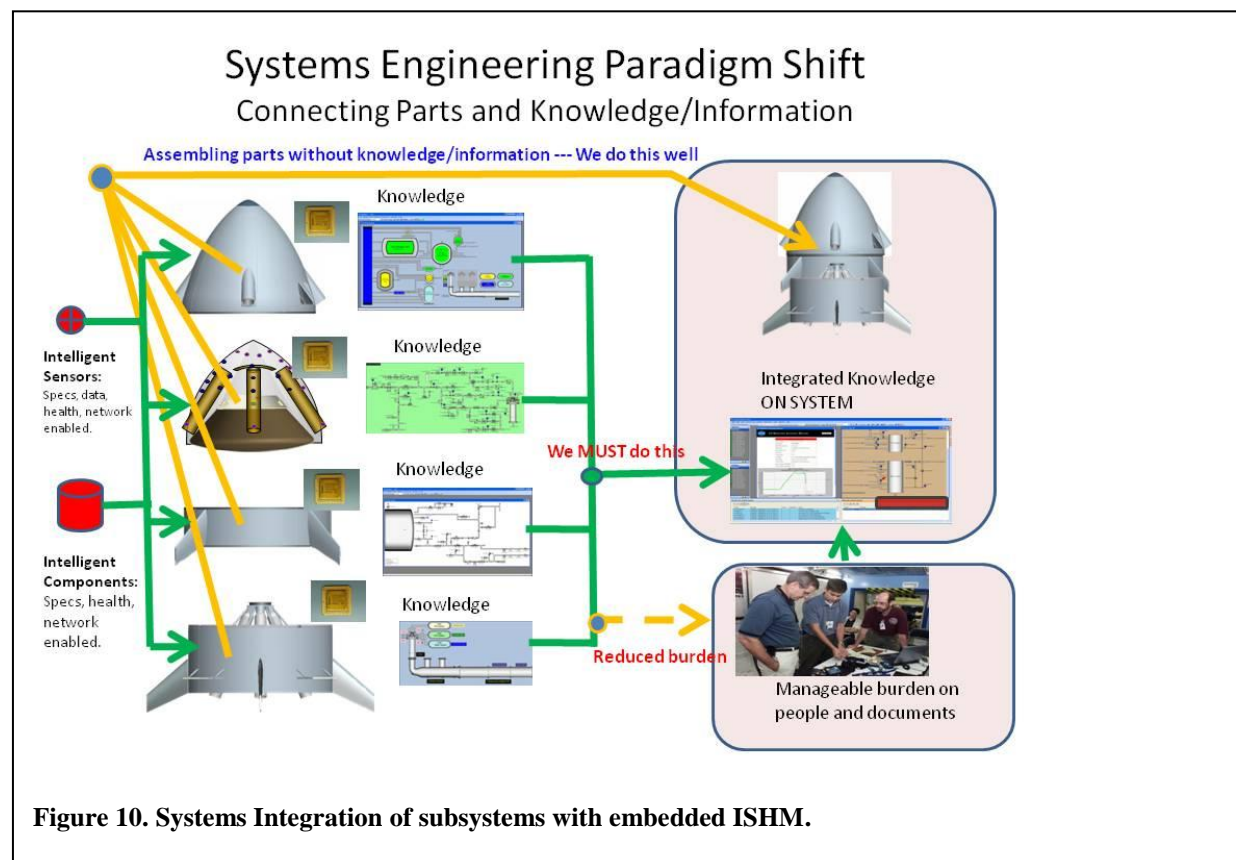
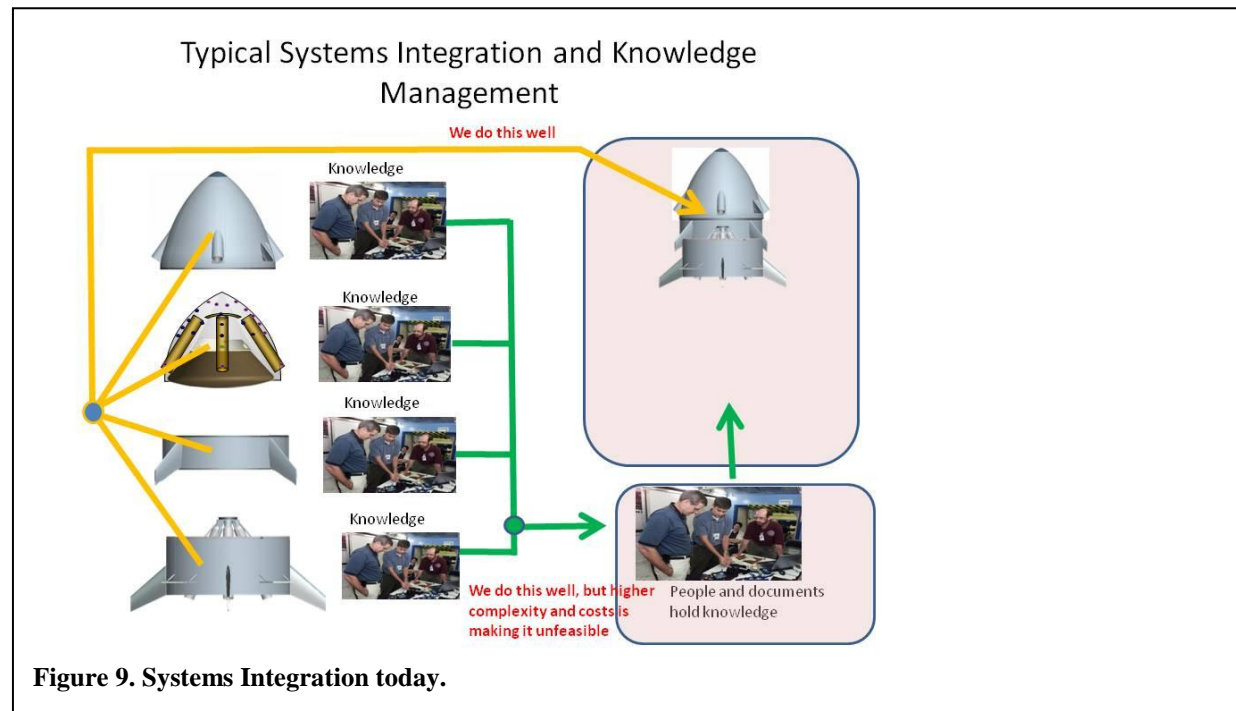


Figure 7. Architecture showing implementation of intelligent sensors through the Virtual Intelligent Sensor Environment.

E. Optimizing sensor selection and placement for ISHM

When developing an ISHM capability from the ground up, one must optimize sensor suites to achieve maximum functional capability (anomaly detection, diagnosis, effects, prognostics). References [18-22] provide context for

- Faster and reliable integration, verification, validation, test, and mission readiness assessment.
- Complete and continues visibility of system condition throughout life-cycle.
- Decreased life-cycle costs.
- Highly self-sufficient systems.
- Efficient evolution to future systems, as one builds upon integrated subsystems with embedded knowledge.



V. Intelligent Control for ISHM-Enabled Systems

Control of complex systems that are ISHM-enabled is a nascent area, simply because ISHM itself is also relatively new. The objective is for the control function to make use of system health information in order to achieve its objectives. Suspect (i.e., disqualified) sensors might be removed from use by critical control functions; anomalous components might need to be contained in order to maintain system function; and, in severe cases, new mission objectives may need to be identified. The paradigm implies that control systems become users of health information, while at the same time making use of actuators to help further improve determination of the system health. This can lead to yet another area of control, specifically focused on helping the ISHM capability detect anomalies, diagnose causes, and determine effects. An example of a control system that incorporates sensor and actuator health information communicated using the IEEE 1451.1 Standard is described in reference [23].

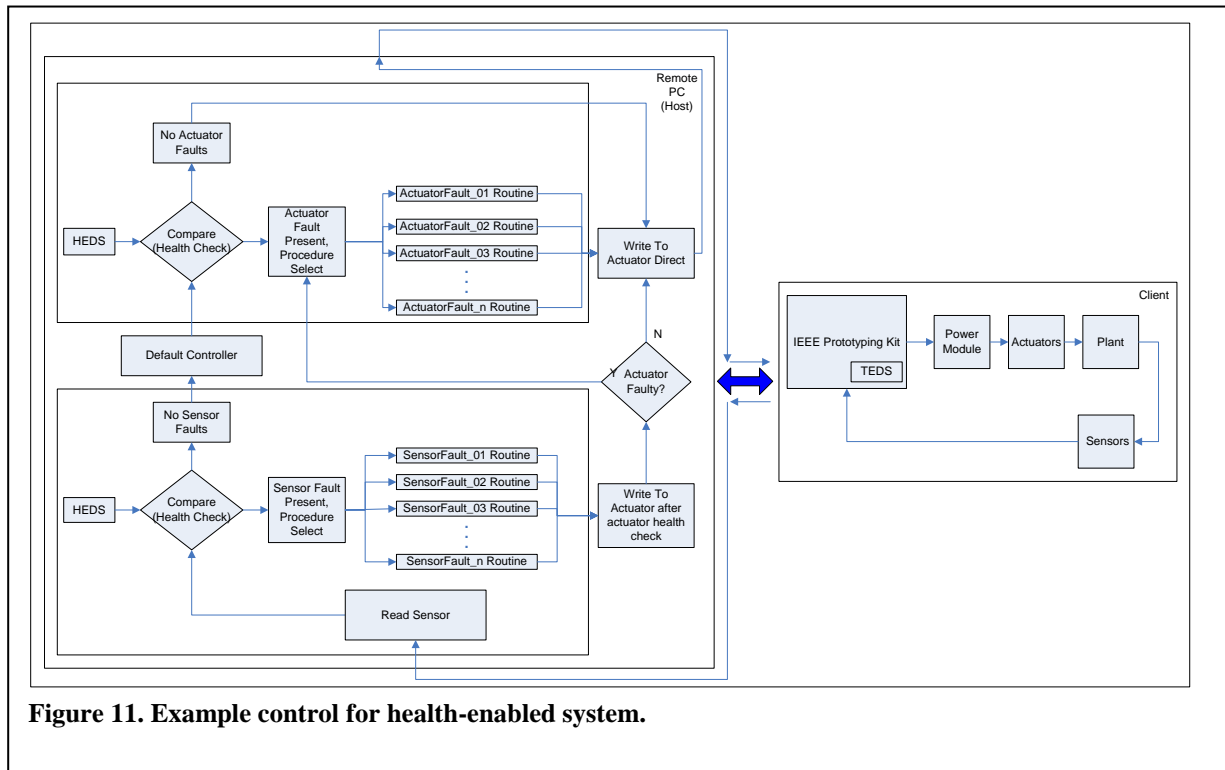


Figure 11. Example control for health-enabled system.

Figure 11 shows a system diagram for a control strategy experiment that incorporates sensor and actuator health, where two key standards in the IEEE 1451 family of standards for smart sensors and actuators were used [23]. The standards implemented include the Transducer Electronic Data Sheet (for automatic identification and specifications), and the NCAP (Network Capable Application Program) used to communicate with the intelligent sensor. The control strategy selects algorithms depending on what faults sensors and/or actuators might exhibit.

VI. Opportunities and Need for Advances in Verification and Validation

The entire chapter essentially describes ISHM capability implementation as purposeful management of DIaK with a focus on determining the health of each element in a system. The need to use knowledge, and hence inference engines; and the complexities of parallel processing and reconciliation of potentially inconsistent outcomes that lead to anomaly determination; requires advances in verification and validation of the ISHM capability itself. This manuscript only raises this issue, but the scope of this topic is broad, it is generic to knowledge-based systems, and is left to be addressed by other colleagues.

VII. Implementation Example: Rocket Engine Test Facility and Test Article

A core pilot ISHM capability implementation was done for a rocket engine test stand and its test article (chemical steam generator) at NASA Stennis Space Center, MS. Details are provided in reference [11]. Multiple objectives were achieved that incorporate many of the technologies, tools, and capabilities discussed in this chapter, but the most significant outcome was to achieve an implementation on an operational system, and use it in real-time, during operations.

The implementation embodied a physical systems architecture that is shown in Figure 7. Intelligent sensor functionality was achieved using the Virtual Intelligent Sensor Environment (VISE). This environment is able to process real-time data streamed from the data bus in the Test Control Center (TCC), or use historical data from files. In this case, real-time meant processing approximately 300 sensors/signals streaming at a rate of 250 samples/sec. The VISE transforms every sensor and signal from the facility into its “intelligent” version. That is, it includes a Transducer Electronic Data Sheet (TEDS) for each sensor/signal; processes data streams to capture anomaly indicators, checks for exceedances on specification limits, and streams data to a bus using the IEEE 1451.1 standard for communications by smart sensors and actuators or NCAP (Network Capable Application Process). Data and sensor health information is also stored in the Health Assessment Database System (HADS) for analysis off-line. Intelligent sensor information (measurement plus health variables) are consumed by the ISHM-DM of the system, running in the ISHM (G2) computer. The VISE’s software architecture is modular, and can systematically be configured to accommodate more sensors and incorporate additional processes that operate on the measurement streams.

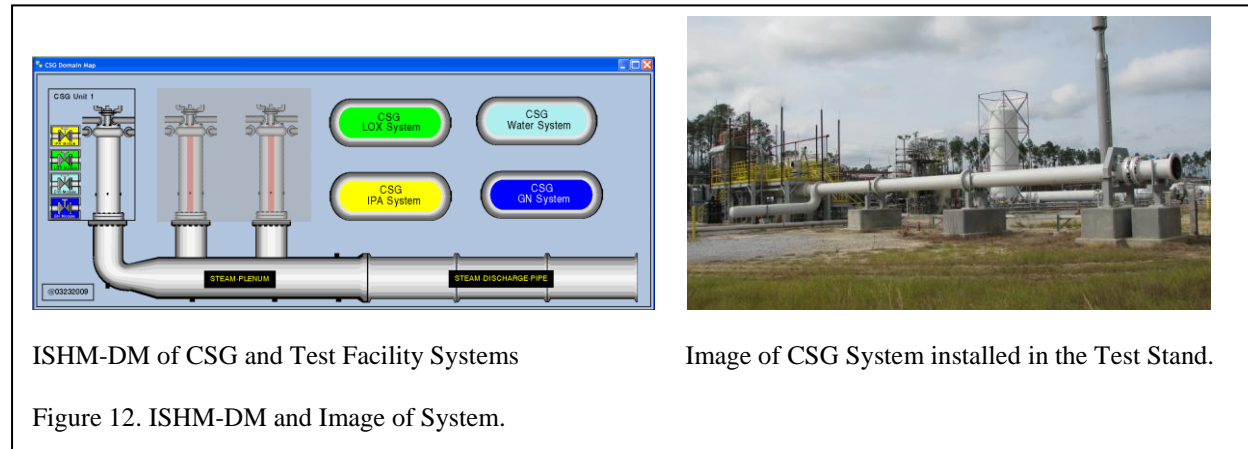


Figure 12. ISHM-DM and Image of System.

Image of CSG System installed in the Test Stand.

Figure 12. ISHM-DM and Image of System.

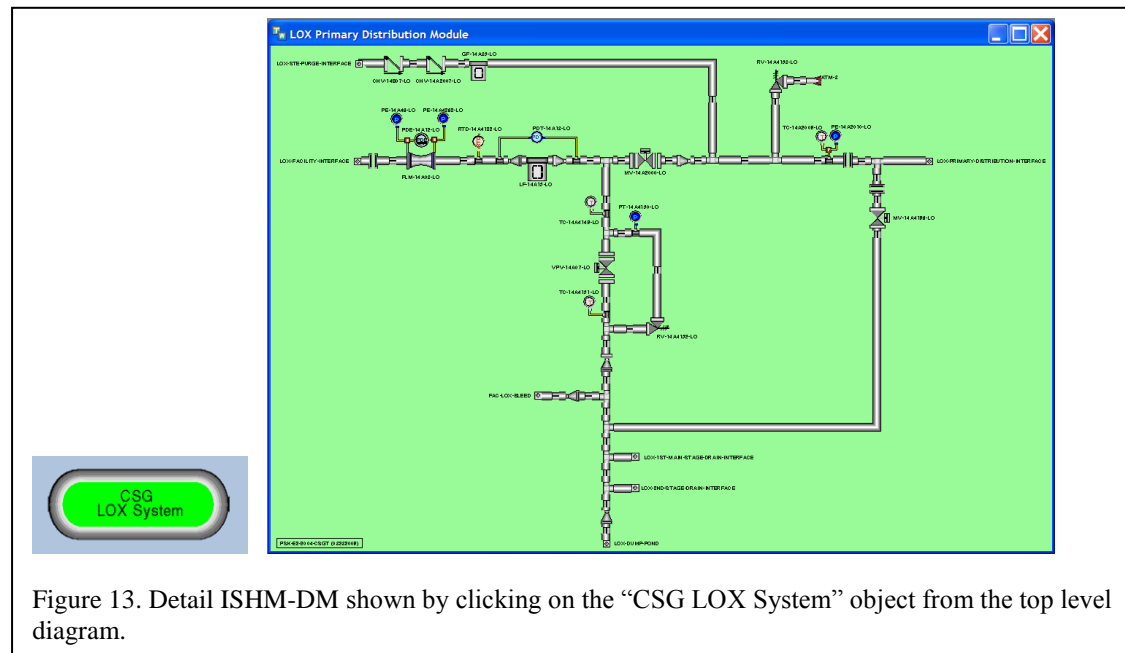


Figure 13. Detail ISHM-DM shown by clicking on the “CSG LOX System” object from the top level diagram.

The ISHM-DM was developed using a toolkit built jointly by NASA and General Atomics (San Diego, CA) in the G2 environment [14]. A description of the General Atomics version is provided in Reference [12]. The DM includes subsystems of the test stand (ovals), subsystems that feed the test article (shown on left of CSG Unit 1), and the test article itself (CSG units, where only one is active to satisfy initial objectives of the test program). Figure 12 shows a top view of the ISHM-DM (left), and an image of the system (right). Each entity in the ISHM-DM represents an object, and any system is a collection of interconnected objects, derived directly from schematic diagrams. Figure 13 shows details of the CSG LOX System displaying interconnected objects directly translated from the system schematic. Each element (pipe, elbow, valve, sensor, etc.) is an “intelligent” object that incorporates information describing who it is (ID and TEDS), what it is (class of object ... such as a valve), and what it can do (parameters relevant to process models where the objects partake, for example operational limits from TEDS or component specifications, or potential failure modes). Figure 14 shows a top level DM window along with sub-windows with lists of objects, sensor TEDS, redline/blueline warning and occurrence lists. .

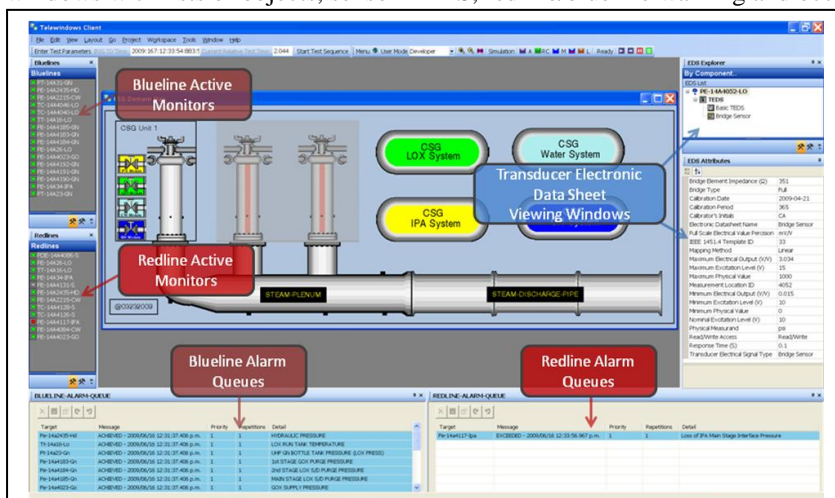


Figure14. An interface screen showing a top view of the ISHM-DM, along with windows containing information regarding TEDS and alarms

When creating an ISHM-DM, the software automatically generates a knowledge base (KB). Objects are selected from a library in the toolkit, dropped in a workspace, and connected to reflect the schematic diagram. The KB generates configuration information derived from interconnections (what is connected to what) made at the moment objects are created and connected. These connection relationships are available for reasoning that might be done with multiple tools typical of object-oriented environments with an inference engine: procedures, methods, rules, and root-cause trees. Class membership

relationships are also inherent in the object classes; for example, a temperature sensor is a member of the higher level sensor class. Figure 15 shows a generic object class architecture suitable for creating ISHM-DM's. Multiple-inheritance can enable incorporating various categories of information for each object, e.g. specifications, process models associations, principles of operation, etc.

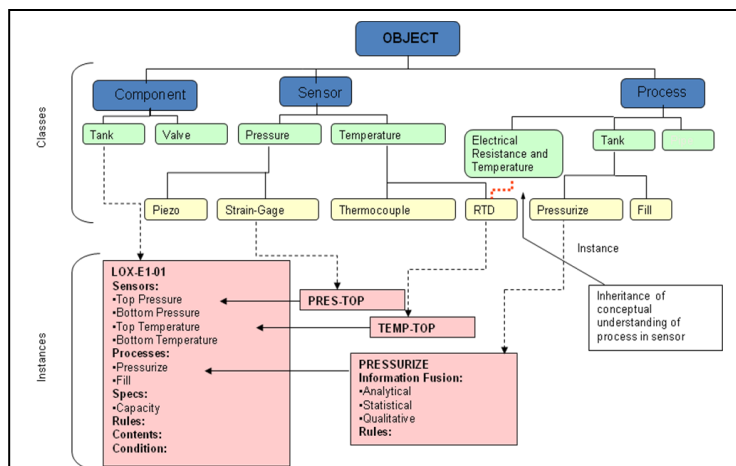


Figure 15. Example Object Class Definitions for ISHM-DM's

An example that illustrates forcefully the need for a knowledge-based ISHM-DM is the implementation of a strategy to detect leaks in isolated subsystems. This strategy is a common sense method of checking for leaks by operators. The condition for the leak check is to identify isolated subsystems that, by definition, should maintain pressure levels. Then identify pressure sensors in the subsystem and check if pressure is maintained. If not, then the subsystem is leaking. That means that, in first instance, all member elements of the subsystem are suspect of leak, creating an ambiguity group. Additional information is used to reduce the ambiguity to a minimum number of suspect or confirmed leak sources. Figure 16 shows the steps as implemented within the ISHM-

DM. At system initialization, a procedure searches (navigates) through pipe objects, noting valves or other isolation elements. Note that all the necessary information for this process is part of the class object definitions in the ISHM-

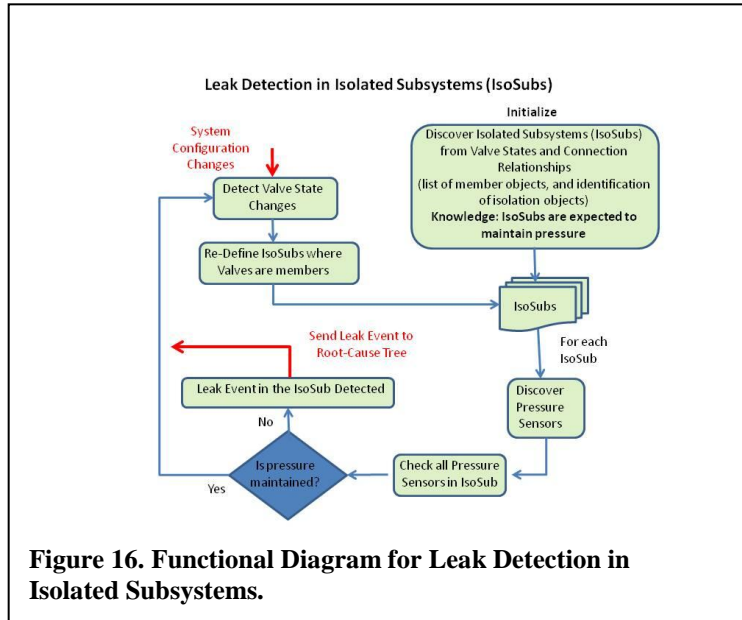


Figure 16. Functional Diagram for Leak Detection in Isolated Subsystems.

procedure enters a monitoring/checking loop pertaining to any valve (or isolation element) configuration changes. If any occur, then only those IsoSubs affected by the valve with changed configuration are analyzed to determine deleted and newly formed IsoSubs. After initialization, the loop runs on a time schedule, but adjusting the IsoSubs occurs every time a valve (or isolation object) change configuration.

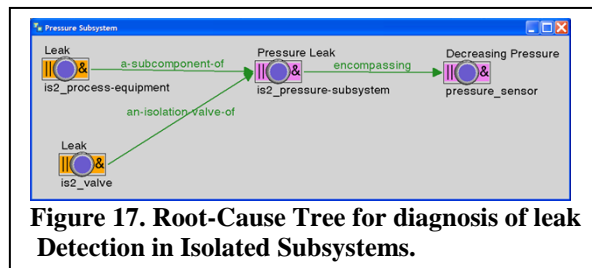


Figure 17. Root-Cause Tree for diagnosis of leak Detection in Isolated Subsystems.

When the procedure detects a leak event in the IsoSub, it is sent to the leak for IsoSubs node of the generic root-cause tree, and instantiated for the particular members of the leaking IsoSub. Figure 17 shows a generic tree for determining causes and effects of leak. The diagram constitutes the code itself; which is very expressive, drawing from information embedded in objects in the ISHM-DM. The tree indicates that a leak event in any is2_flow-subsystem (what has been called an IsoSub) can be caused by leaks on any is2_process-equipment (any element) that is a-subcomponent of the IsoSub, but also treats separately a subcomponent that is an isolation valve. The reason is that isolation valves belong to adjacent IsoSubs, and if the adjacent IsoSub is not leaking, then one can immediately conclude that the valve is not leaking, thus reducing the size of the ambiguity group that may be causing the

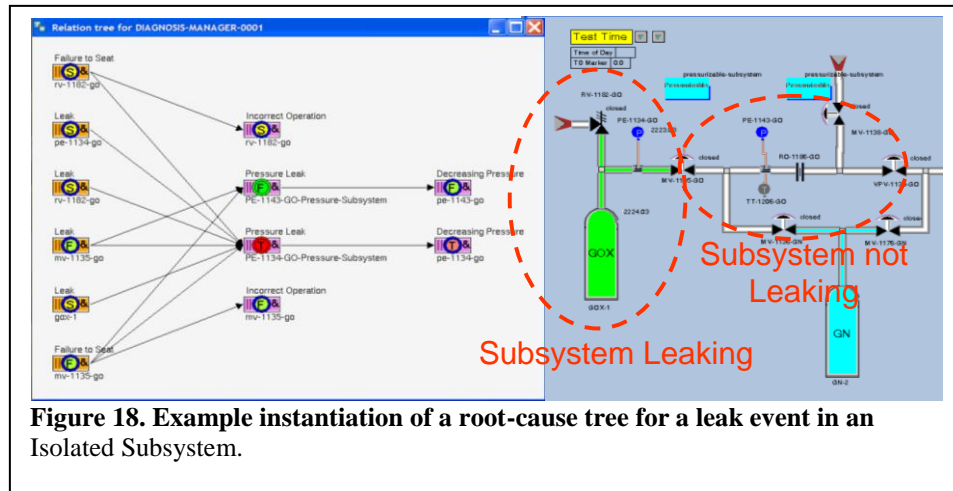


Figure 18. Example instantiation of a root-cause tree for a leak event in an Isolated Subsystem.

leak.

Figure 18 shows the occurrence of an instantiation of a leak event in an IsoSub. The right side shows a diagram (ISHM-DM) of a gaseous oxygen (GOX) and gaseous nitrogen (GN) subsystems of a simple experimental rocket engine test stand. The leak detection procedure has identified all IsoSubs, including the two marked within the red oval spaces. Pressure decrease in the sensor on the left has created a leak event in that subsystem. The event has been sent to the generic root-cause for leak (Figure 17), and instantiated for that subsystem. When that happens, a detailed cause-effect diagram involving all elements in the subsystem is automatically created. Nodes for each subcomponent of the leaking IsoSub are overlaid with indicators that describe them as suspect (S), false (F), true (T), and also if the indication is inferred or direct. Note that the node corresponding to the valve isolating the two IsoSubs has an F indicator, meaning that it is not leaking. The other node for the valve also indicates that it is false that the valve has an anomaly called “failure to seat.” This conclusion comes from other root-cause trees describing failure modes of valves.

VIII. Conclusion

This paper describes concepts, architectures, paradigms, tools, and implementations of ISHM capability. The purpose is to show that ISHM capability must be implemented as a Knowledge-based capability, through management of data, information, and knowledge (DIaK); whereby “management” implies storage, distribution, sharing, maintenance, processing, reasoning, and presentation. The emphasis is also to note that ISHM capability increments “intelligence” of the system where it is implemented. We can then talk about ISHM-Enabled systems; with a potential to generate significant advances in Systems Design, Integration, and Engineering; as well as in Systems Control. The reader should also infer that much work is needed in developing ISHM-DM’s, tools to create and use the ISHM-DM’s, and implementation of standards for management of DIaK to achieve Plug&Play and interoperability. The concept of ISHM-DM encompasses an integrating knowledge model about the entire system, specially incorporating knowledge that makes possible automating interactions and analysis, and use in reasoning and decision making. Last, but not least, it is important to note that the ISHM DIaK Architecture (DIaKA) described addresses the need for focusing on processes that take place in systems for consistency checking, leading to anomaly detection.

Acknowledgements

The authors would like to thank NASA for providing the opportunity to work on advancing the area of ISHM. The authors also express their profound appreciation to the many individuals that through discussions and interactions have enriched their understanding of ISHM and made possible this chapter.

Bibliography

1. Pamela Surko and June F. Zakrajsec, “PTDS: Space Shuttle Main Engine Post Test Diagnostic Expert System for Turbopump Condition Monitoring,” SAE Technical Paper Series 922059, Aerotech ’92, Anaheim, CA, October 5-8, 1992.
2. Ramohalli, G., 1994, “Honeywell’s Aircraft Monitoring and Diagnostic Systems for the Boeing 777,” *Proceedings of the 17th Symposium on Aircraft Integrated Monitoring Systems*, pp. 69-71, 73-87.
3. Davidson, M. and Stephens, J., 2004, “Advanced Health Management System for the Space Shuttle Main Engine,” *Proceedings 40th AIAE/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, AIAA 2004-3912.
4. PITEX reference xxxx.
5. Kang B. Lee, “Smart Transducer Interface Standards for Condition Monitoring and Control of Machines,” *Condition Monitoring and Control for Intelligent Manufacturing*, edited by L. Wang and R. Gao, Springer Series in Advanced Manufacturing, Springer Verlag, UK, 2006, pp. 347-372.
6. http://www.nist.gov/manuscript-publication-search.cfm?pub_id=821920
7. <http://www.mimosa.org>.
8. Karl Reichard, Fernando Figueroa, Rebecca Oosdyke, John Schmalzel, and Jose Perotti, “An ISHM Architecture for Ground Operations Health Management,” ISHM Conference, Covington Convention Center, KY, August 11-14, 2008.
9. Figueroa, F., Schmalzel, J., Walker, M., Venkatesh, M., Kapadia, R., “Integrated System Health Management: Foundational Concepts, Approach, and Implementation,” *AIAA Infotech@Aerospace Conference*, 20-22 April 2010.
10. Figueroa, F., and Schmalzel, J., “Rocket Testing and Integrated System Health Management,” *Condition Monitoring and Control for Intelligent Manufacturing*, edited by L. Wang and R. Gao, Springer Series in Advanced Manufacturing, Springer Verlag, UK, 2006, pp. 373-392.
11. Fernando Figueroa, John Schmalzel, Jonathan Morris, Mark Turowski, and Richard Franzl, “Integrated System Health Management: Pilot Operational Implementation in a Rocket Engine Test Stand,” *AIAA 2010-3454*, AIAA Infotech@Aerospace 2010 Conference, 20-22 April 2010, Atlanta, GA.
12. R. Kapadia, R. Gross, M. Walker, M. Venkatesh. “Health Monitoring Assessment and Prognostics (HealthMAP™) for Advanced Arresting Gear System. PHM Society Conference 2009, San Diego, CA, October 2009.

13. M. Walker, "Model-based Reasoning Applications for Remote Intelligent Systems Health Management", ASNE Intelligent Ships Symposium, May 2007.
14. Gensym (<http://www.gensym.com>).
15. Esensors, Inc. (<http://www.eesensors.com/index.html>).
16. Mobitrum Corporation (<http://www.mobitrum.com>).
17. National Instruments (<http://www.ni.com>).
18. Melcher, K.J., Sowers, T.J., Maul, W.A., "Meeting the Challenges of Exploration Systems: Health Management Technologies for Aerospace Systems with Emphasis on Propulsion," NASA TM 2005-214026, *First International Forum on Integrated System Health Engineering and Management in Aerospace*, 7-10 November 2005.
19. Santi, L.M., Sowers, T.S., Aguilar, R.B., "Optimal Sensor Selection for Health Monitoring Systems", AIAA 2005-4485, *AIAA 41st Joint Propulsion Conference and Exhibit*, 10-13 July 2005.
20. Maul, W.A., Melcher, K.J., Chicatelli, A.K., and Sowers, T.S., "Sensor Data Qualification for Autonomous Operation of Space Systems," NASA TM 2006-214475, 2006 Fall Symposium Series sponsored by the American Association for Artificial Intelligence, 13-15 October 2006.
21. Melcher, K.J., Fulton, C.E., Maul, W.A., Sowers, T.S., "Development and Application of a Portable Health Algorithms Test System," NASA TM-2007-214840, 54th Joint Army-Navy-NASA-Air Force (JANNAF) Propulsion Meeting, 14-17 May 2007.
22. Melcher, K.J., Maul, W.A., Fulton, C.F., Wong, E., "Sensor Data Qualification Proof-of-Concept Demonstration," NASA TM-215518, Joint Army-Navy-NASA-Air Force (JANNAF) 6th Modeling & Simulation Subcommittee, 4th Liquid Propulsion Subcommittee, 3rd Spacecraft Propulsion Subcommittee Joint Meeting, 8-12 December 2008.
23. D. Jethwa, R. R. Selmic and F. Figueroa, "Real-time implementation of intelligent actuator control with a transducer health monitoring capability," *International Journal of Factory Automation, Robotics, and Soft Computing*, no. 1, pp. 5-10, January 2009.
24. Bajwa, A. and Sweet, A., 2002, "The Livingstone Model of a Main Propulsion System," *RIACS Technical Report 03.04*. Available at <http://www.riacs.edu/trs/>.
25. Qualtech Systems Inc. (<http://www.teamqsi.com/index.html>).
26. PHM Technology (<http://www.phmtechnology.com/>).